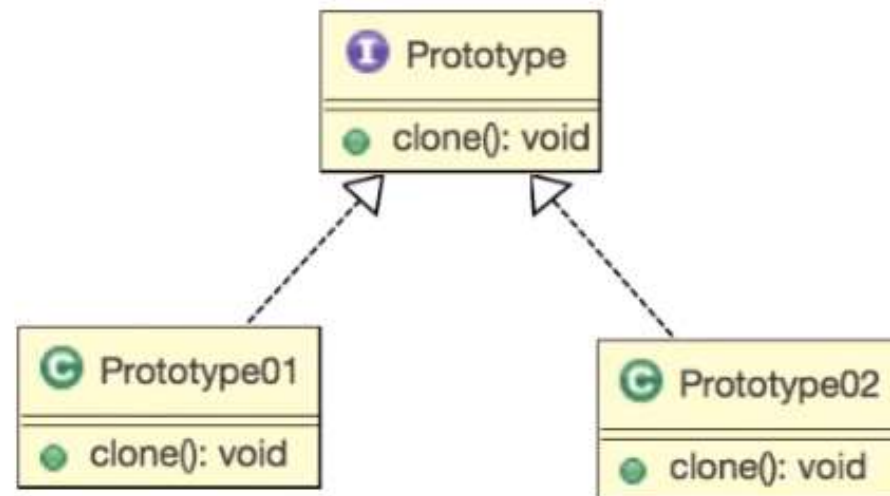


Prototype Pattern

2019.02.27.

Goal of Lecture

Duplicate Complex Instance through the Prototype Pattern



Concept

Prototype Pattern: Easily Create through duplicate what Instance required high Production cost

Case of High Production cost

- i) There are too many classes so they can not be organized into classes.
- ii) Instance creation from class is difficult.

Requirement

Develop Draw Tool Program

Shape.java

```
1 public class Shape implements Cloneable{
2     private String id;
3
4     public void setId(String id){
5         this.id = id;
6     }
7     public String getId(){
8         return id;
9     }
10 }
11
12 }
13 /**
14  * @Override
15  * protected Object clone() throws CloneNotSupportedException{
16  *     return super.clone();
17  * }
18 */
```

Example for this one

check on IDE

Circle.java

Circle x.y.r (

Getter/Setter
x.y.R

Clone ← [

```
1 public class Circle extends Shape {  
2     private int x, y, r;  
3  
4     public Circle(int x, int y, int r) {  
5         super();  
6         this.x = x;  
7         this.y = y;  
8         this.r = r;  
9     }  
10    public int getX() {  
11        return x;  
12    }  
13    public int setX(int x) {  
14        this.x = x;  
15    }  
16    public int getY() {  
17        return y;  
18    }  
19    public int setY(int y) {  
20        this.y = y;  
21    }  
22    public int getR() {  
23        return r;  
24    }  
25    public int setR(int r) {  
26        this.r = r;  
27    }  
28  
29    public Circle copy() throws CloneNotSupportedException {  
30        Circle circle = (Circle) clone();  
31        return circle;  
32    }  
33 }
```

Main.java

```
1  public class Main{
2      public static void main(String[] args){
3          Circle circle1 = new Circle(1, 1, 3);
4          Circle circle2 = circle1.copy();
5
6          System.out.println(circle1.getX()+" "+
7                               circle1.getY()+" "+
8                               circle1.getR());
9
10         System.out.println(circle2.getX()+" "+
11                              circle2.getY()+" "+
12                              circle2.getR());
13     }
14 }
15 }
```

Result


$$\begin{array}{r} 1.1.3 \\ \hline 1.1.3 \end{array}$$

Requirement

I don't want overlapped circles each other.

Add)Circle.java

```
public Circle copy()throws CloneNotSupportedException{  
    Circle circle = (Circle) clone();  
    circle.x = x + 1;  
    circle.y = y + 1;  
    return circle;  
}
```



Next! Shallow Copy and Deep Copy.